

**CSTA STANDARDS CORRELATIONS for CCR CS-1 & CS-2**

Theme	Computer Science-1	Computer Science-2
<p><b>Computing Systems</b> Includes Devices, Hardware/Software, &amp; Troubleshooting</p>	<p>2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-02 Design projects that combine hardware and software components to collect and exchange data. 2-CS-03 Systematically identify and fix problems with computing devices and their components.</p> <p>3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.Z)</p>	<p>3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. 3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. 3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.</p> <p>3B-CS-01 Categorize the roles of operating system software. 3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.</p>
<p><b>Networks &amp; the Internet</b> Includes Network Organization/Communication &amp; Cybersecurity</p>	<p>2-NI-04 Model the role of protocols in transmitting data across networks and the Internet. 2-NI-05 Explain how physical and digital security measures protect electronic information.</p> <p>3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.</p>	<p>3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks. 3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. 3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. 3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.</p> <p>3B-NI-04 Compare ways software developers protect devices and information from unauthorized access.</p>
<p><b>Data &amp; Analysis</b> Includes Storage, Collection/Visualization/Transformation, &amp; Inference/Models</p>		<p>3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. 3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.</p> <p>3B-DA-05 Use data analysis tools and techniques to identify patterns in data representing complex systems.</p>
<p><b>Algorithms &amp; Program Design</b> Includes Algorithms, Variables, Control, Modularity, &amp; Program Development</p>	<p>2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 Systematically test and refine programs using a range of test cases. 2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.</p>	<p>3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but</p>
<p><b>Algorithms &amp; Program Design</b> Includes Algorithms, Variables, Control, Modularity, &amp; Program Development</p>	<p>3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.</p>	<p>3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users. 3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.</p> <p>3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. 3B-AP-10 Use and adapt classic algorithms to solve computational problems. 3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects. 3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. 3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs. 3B-AP-17 Plan and develop programs for broad audiences using a software life cycle process. 3B-AP-18 Explain security issues that might lead to compromised computer programs. 3B-AP-19 Develop programs for multiple computing platforms. 3B-AP-20 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. 3B-AP-24 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.</p>
<p><b>Impacts of Computing</b> Includes Culture, Social Interactions, &amp; Safety/Law/Ethics</p>	<p>2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies. 2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.</p> <p>3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. 3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. 3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.</p>	<p>3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. 3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. 3A-IC-27 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. 3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.</p> <p>3B-IC-27 Predict how computational innovations that have revolutionized aspects of our culture might evolve.</p>

from CSTA: Level 3A standards are recommended for all students before the end of high school; Level 3B standards are for students who wish to pursue a CS career or education beyond high school.